Unisul 2019

Aluno: Gabriel Cavalcante dos Santos

Matéria: Análise de Algoritmos

Problema do Troco

Um dos mais clássicos problemas de ciência da computação é o problema do troco em moedas. O problema em si existe desde que os seres humanos começaram a fazer trocas voluntárias, e na era da computação, esse problema parece cada vez mais secundário, mas sua resolução ainda tem muito sentido e valor por questões de aprendizado.

A questão foi primeiramente observada no campo da economia, e a medida que a computação foi tomando conta das atividades rotineiras e repetitivas da vida humana, esse problema foi trazido para ser contemplado no campo da ciência da computação.

O problema do troco (no inglês, coin change problem) consiste em encontrar uma combinação com o menor número de moedas cuja soma seja igual a uma quantia determinada, partindo da premissa de que há uma lista de moedas válidas que possuem disponibilidade infinita. Levando em conta as denominações padrões da moeda brasileira, o Real, há moedas para R$0,01, R$0,05, R$0,10, R$0,25, R$0,50 e R$1,00. Na prática, o objetivo que leva a resolução do problema é, dada uma certa quantia (geralmente trabalha-se com um número inteiro de centavos), encontrar um caminho de instruções no algoritmo que fornece a quantia com uma quantidade de moedas válidas mínima. Em termos de programação, esta quantidade de moedas geralmente é tida como infinita. Isso faz com que, no programa, para qualquer quantia de centavos necessários para o troco, há uma solução possível, independente do caixa real.

É importante notar que o problema apresentado é levemente similar a outro problema clássico da computação; o problema da mochila, em que se busca colocar produtos que maximizam o valor de uma mochila com uma restrição, geralmente peso ou espaço. De certa forma, o problema do troco é uma inversão do problema da mochila: com um valor fixo para a mochila, é necessário encontrar a combinação de produtos com peso menor que fornece esse valor.

Programação dinâmica

Uma das abordagens mais comuns utilizadas para resolver o problema se dá através dos algoritmos ditos “gulosos”. Este tipo de algoritmo ira buscar apenas a moeda que mais se aproxima do resultado, independente do número de moedas utilizadas na solução. Isso faz com que, em algumas circunstâncias, o método guloso não seja o método de resolução que apresente a solução ótima, ou seja, não é o algoritmo que utiliza a menor quantidade de moedas para chegar ao resultado.

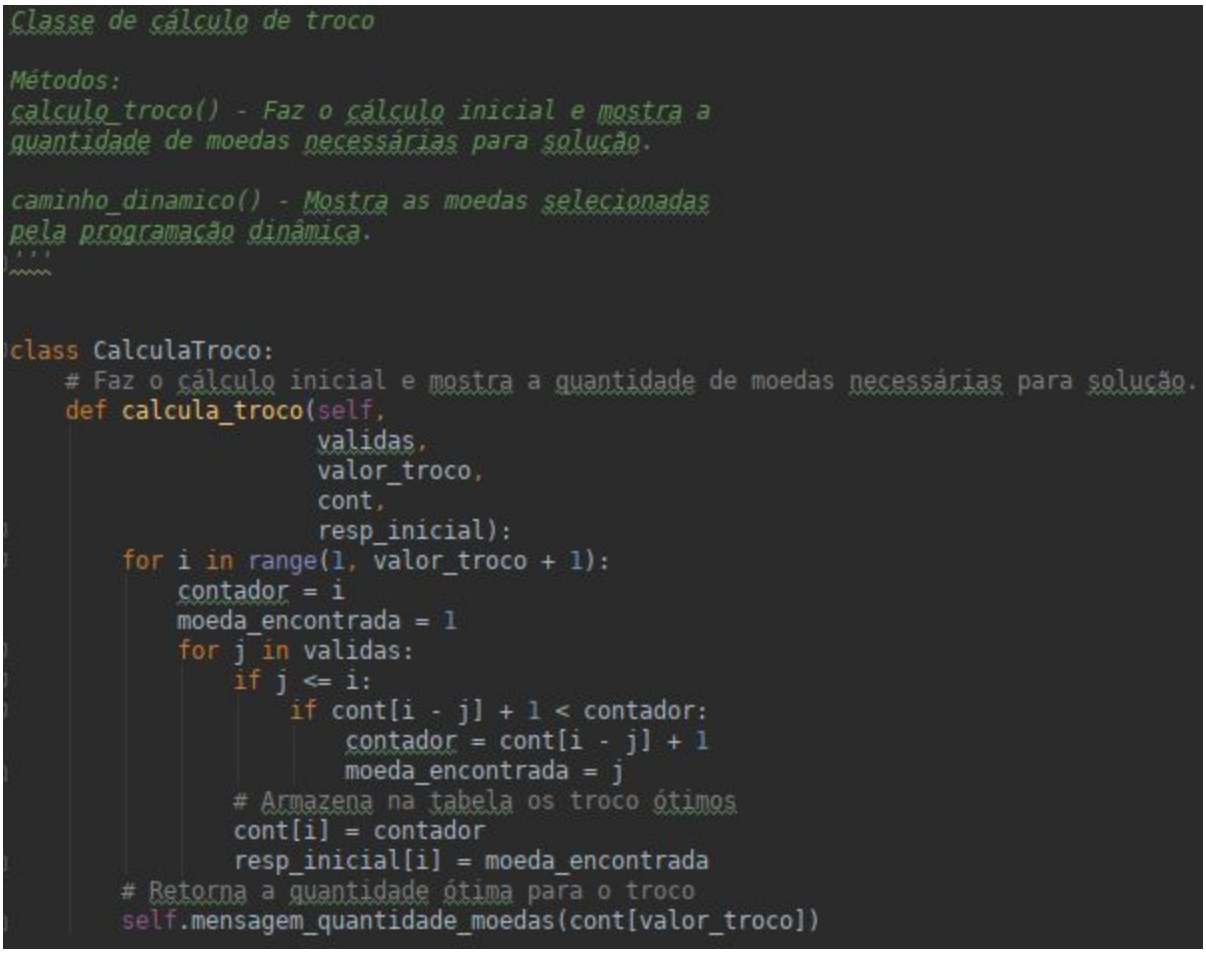
Uma solução que chega mais próxima da requirida pela descrição do problema é a utilização da programação dinâmica, que busca o resultado ótimo de forma diferente, otimizando a recompensa, conhecida na programação também como “função objetivo”, para alcançar uma “visão geral” da solução, em vez de buscar apenas o ganho imediato.

A solução proposta para o problema, utilizando programação dinâmica, envolve desenvolver um algoritmo que, a partir de uma lista de k moedas válidas, encontre qual moeda pode ser escolhida para se encontrar o caminho ótimo que leva até o valor do troco, e repetir esse processo até que o valor total das moedas escolhidas seja o valor total do troco.

Algoritmo

O problema do troco é apresentado, na prática, neste texto, na linguagem Python. Utilizando a programação orientada a objeto (POO), o projeto é dividido em classes, uma classe para lidar com os métodos relacionados diretamente com o algoritmo do problema (Troco.py), e outra classe usada para colocar todos os métodos em prática usando parâmetros definidos nela mesma (Principal.py). O programa pode ser iniciado executando o arquivo ‘Principal.py’.

A imagem abaixo mostra a primeira parte da classe ‘CalculaTroco’, responsável pelo algoritmo apresentado como solução para o problema proposto.



```
'''Classe de cálculo de troco

Métodos:
calculo_troco() - Faz o cálculo inicial e mostra a
quantidade de moedas necessárias para solução.

caminho_dinamico() - Mostra as moedas selecionadas
pela programação dinâmica.
'''


class CalculaTroco:
    # Faz o cálculo inicial e mostra a quantidade de moedas necessárias para solução.
    def calcula_troco(self,
                      validas,
                      valor_troco,
                      cont,
                      resp_inicial):
        for i in range(1, valor_troco + 1):
            contador = i
            moeda_encontrada = 1
            for j in validas:
                if j <= i:
                    if cont[i - j] + 1 < contador:
                        contador = cont[i - j] + 1
                        moeda_encontrada = j
                # Armazena na tabela os troco ótimos
                cont[i] = contador
                resp_inicial[i] = moeda_encontrada
        # Retorna a quantidade ótima para o troco
        self.mensagem_quantidade_moedas(cont[valor_troco])
```

A função ‘calculo_troco’ faz o cálculo inicial para mostra quantas moedas são necessárias para uma determinada quantia de troco.

A segunda metade da classe diz respeito especificamente ao uso da programação dinâmica para endereçar o problema.

```
    def caminho_dinamico(self,
                         resp_inicial,
                         valor_troco):
        moedas = []

        while valor_troco > 0:
            moeda_escolhida = resp_inicial[valor_troco]
            moedas.append(moeda_escolhida)
            valor_troco = valor_troco - moeda_escolhida

        # Mostra as moedas selecionadas para o método dinâmico
        self.mensagem_resultados(moedas)

    def mensagem_quantidade_moedas(self,
                                   quant):
        print("\nA quantidade de moedas necessárias para o resultado são: {}.\n".format(quant))

    def mensagem_resultados(self,
                            moedas):
        print("Essas são as moedas escolhidas para o troco:\n{}.\n".format(moedas))

    # Método para mostrar a quantia do troco.
    def mensagem_quantia(self,
                         quantia):
        print("O troco necessário para conlcuir essa trasação, em centavos, é de {} centavos.\n".format(quantia))

# Fim do algoritmo
```

O método ‘caminho_dinâmico’ mostra todas as moedas utilizadas na solução.

Os métodos de mensagem são utilizados nos dois métodos previamente mencionados e mostram os resultados dos dois algoritmos.

Na classe ‘Principal’, o algoritmo é colocado em execução juntando todos os métodos da classe anterior. A classe em si contém os parâmetros necessários para a execução de todo o programa.

```python
from codigo.Troco import CalculaTroco

'''
Classe principal

Classe onde os métodos da classe Troco são utilizados.
'''


class Principal:
    # Variável para a nova instância de CalculaTroco()
    cal = CalculaTroco()

    # Valor do troco.
    valor_troco = 49

    # Lista de moedas disponíveis. Baseada no Real (R$)
    moedas_disponiveis = [100, 50, 25, 10, 5, 1]
    resp_incial = [0] * (valor_troco + 1)
    contador = [0] * (valor_troco + 1)

    # Mostra a valor do troco em centavos.
    cal.mensagem_quantia(valor_troco)

    # Faz o cálculo inicial e mostra a quantidade de moedas.
    cal.calcula_troco(moedas_disponiveis, valor_troco, contador, resp_incial)

    # Mostra as moedas utilizando a programação dinâmica.
    cal.caminho_dinamico(resp_incial, valor_troco)
```

Os resultados podem ser vistos no terminal. São mostradas as moedas, a quantidade de moedas e o valor do troco.

```
/usr/bin/python3.7 /home/plebeu/git/ProblemaTroco/ProblemaTrocoPython/codigo/Principal.py
O troco necessário para conlcuir essa trasação, em centavos, é de 49 centavos.


A quantidade de moedas necessárias para o resultado são: 7.

Essas são as moedas escolhidas para o troco:
[25, 10, 10, 1, 1, 1, 1].
```